A LAB REPORT ON ONLINE QUIZ MANAGEMENT SYSTEM

SOHARDYA GOLDER | ID: 201006412 | CSE 124.3

FALL 2023 | DECEMBER 21, 2023

Online Quiz Management System

Introduction:

The Online Quiz Management System is built as a Java application using the Eclipse IDE. It incorporates the principles of object-oriented programming (OOP). This system enables users to create, manage and take quizzes providing a user interface, for both quiz administrators and participants.

1. Question Class:

The Question class is designed to hold a quiz question. It has attributes, like "question Text" to store the question "options", for multiple choice answers and "correct Option" to indicate the answer.

2. Quiz Class:

The Quiz class is responsible, for handling a group of Question instances. It offers functions to add and remove questions making it easy to create quizzes on the fly. This class also supports quiz related operations ensuring integration, with other components.

3. Player Class:

The Player class is designed to represent a participant in a quiz. It contains attributes such, as the players name and score which provide information about their identity and performance, in the quiz. This class allows for the monitoring of individual player scores.

4. Quiz Manager Class:

The Quiz Manager class serves as the main controller for the application. It orchestrates the flow of the quiz, handles user input, and manages quiz-related data. The start Quiz method initiates the quiz, while display Results presents the final scores and outcomes.

Functionality:

1.Quiz Creation and Management

2.Question Addition

3.Question Removal

4.Player Interaction

5.Player Registration

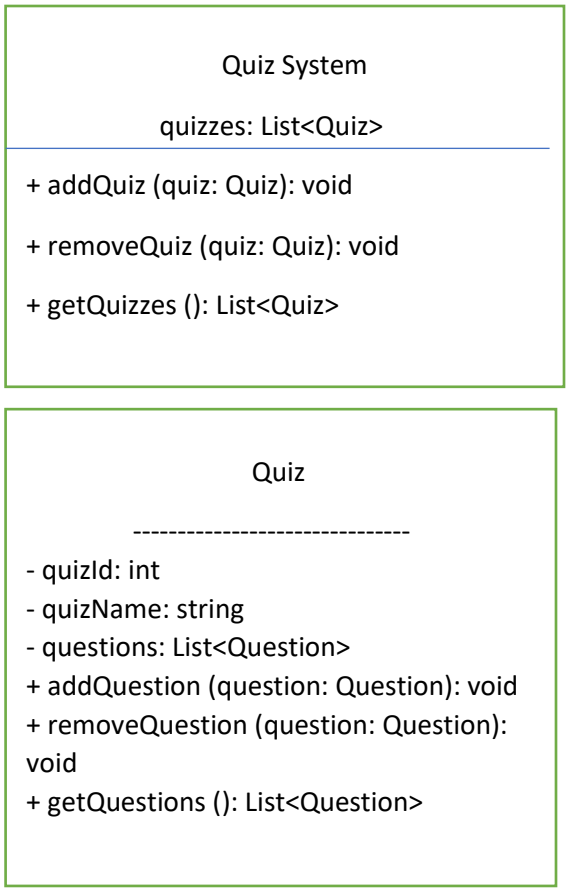
6.Quiz Execution

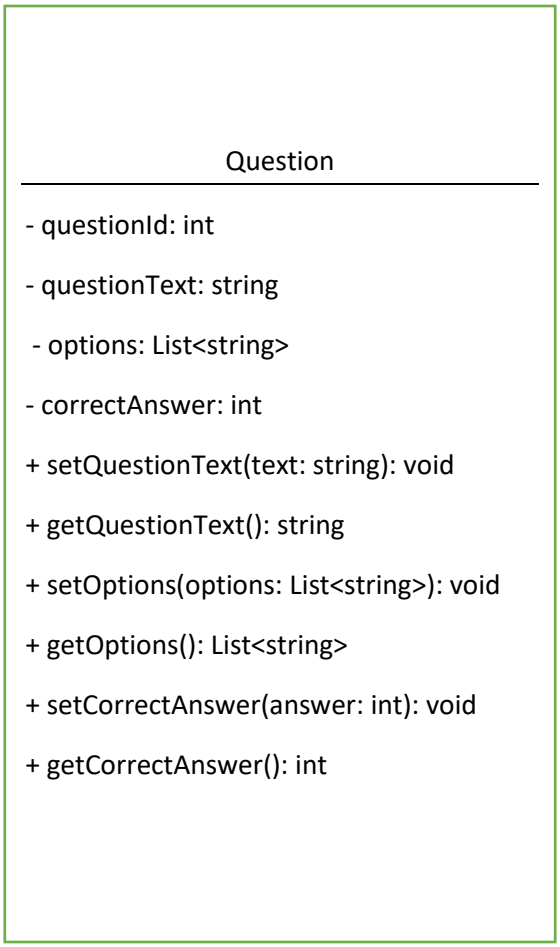
7.Results Display

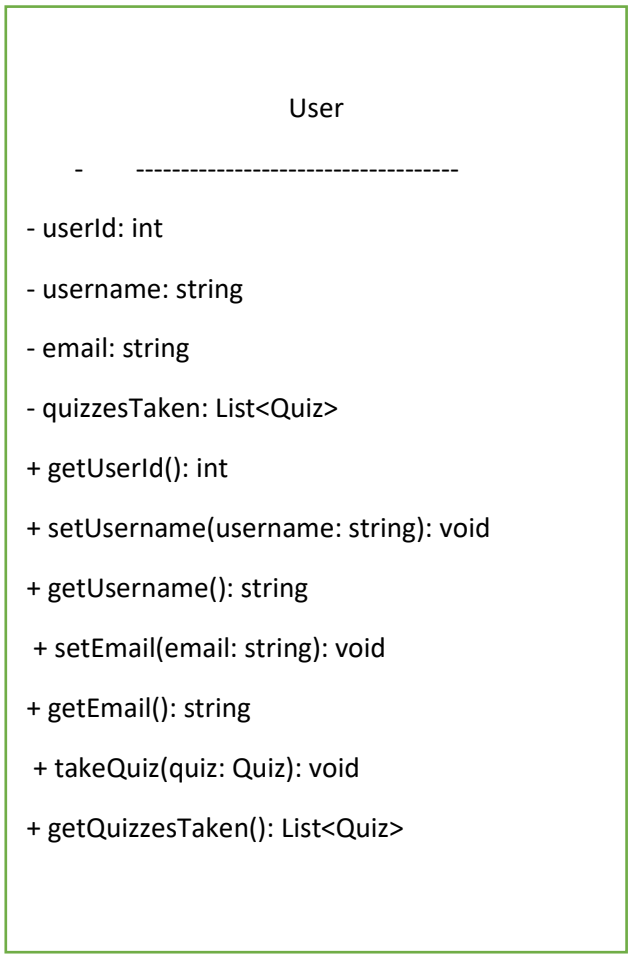
8.Score Calculation

Once the quiz is finished the system will display the results, which include information, about the scorer and detailed performances of each player.

Class Diagram:







Source Code:

1.Question Class:

User

```
public class Question {
    private String questionText;
    private String[] options;
    private int correctOption;
   }
```

2.Quiz Class:

```
import java.util.ArrayList;
  import java.util.List;

  public class Quiz {
     private List<Question> questions;

  }
```

3.User/Player Class:

```
public class Player {
     private String playerName;
     private int score;
  }
```

4.QuizManager Class:

```
import java.util.Scanner;

  public class QuizManager {
     private Quiz quiz;
     private List<Player> players;

     public QuizManager() {
        this.quiz = new Quiz();
        this.players = new ArrayList<>();
      }
```

```
    public void startQuiz() {
    }

    public void displayResults() {
    }

    public static void main(String[] args) {
        QuizManager quizManager = new QuizManager();
        quizManager.startQuiz();
        quizManager.displayResults();
    }
}
```

Explanations:

1. Question Class:

This class represents a single question in the quiz. Here-

- questionText: Contains the text of the question.
- options: An array holding the different options for the question.
- correctOption: Represents the index of the correct option in the options array.

2. Quiz Class:

This class represents the quiz itself.

- questions: A list containing instances of the Question class. It holds multiple questions for the quiz.

3. Player Class:

- This class represents a player/user participating in the quiz.
- playerName: Represents the name of the player.
- score: Indicates the player's score in the quiz.

4. QuizManager Class:

This class manages the quiz-related functionalities.

- quiz: Represents the quiz object.
- players: Maintains a list of Player objects.
- startQuiz(): A method that should handle the flow of the quiz, including presenting questions, receiving user input for answers, calculating scores, etc.
- displayResults(): A method to display the results of the quiz, showing players' scores, and any other relevant information.
- main(): The entry point of the application. It creates an instance of QuizManager, starts the quiz, and then displays the results.

Conclusion:

The structure that has been outlined serves, as a foundation for creating a quiz application using Java. It establishes the classes to handle questions, players and the overall quiz process. By incorporating the logic in functions, like startQuiz () and displayResults () this project can be developed further into a functional quiz application that allows multiple players to participate in an engaging quiz experience. To enhance this foundation some additional features can be added to improve user friendly quiz platform.